

ANALISIS PERFORMANSI QUERY MYSQL MENGGUNAKAN QUERY BUILDER PADA FRAMEWORK CODEIGNITER 4

PERFORMANCE ANALYSIS

OF MYSQL QUERY USING QUERY BUILDER CODEIGNITER 4 FRAMEWORK

Zuhar Musliyana¹, Ayu Helinda²

¹Program Studi Sistem Informatika, Fakultas Ilmu Komputer, Universitas Ubudiyah Indonesia

²Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Ubudiyah Indonesia

Jln. Alue Naga Tibang, Syiah Kuala, Kota Banda Aceh 23116

Corresponding Author: zuhar@uui.ac.id

Abstrak— Saat ini kebutuhan akan penggunaan database terus meningkat seiring dengan perkembangan teknologi informasi. Diantara platform database yang paling banyak digunakan saat ini adalah MySQL. Database MySQL menawarkan berbagai kemudahan untuk menangani penyimpanan data dengan jenis yang beragam. Selain itu, database MySQL juga bersifat open source sehingga memiliki banyak pengembang dan mempunyai jumlah pengguna yang banyak jika dibandingkan dengan platform lainnya. Pada implementasinya database MySQL dapat dikelola secara mudah menggunakan Application Programming Interface (API) melalui berbagai jenis bahasa pemrograman salah satunya yaitu PHP. Codeigniter merupakan salah satu jenis framework yang ditulis dengan bahasa pemrograman PHP. Pada tahun 2020 Codeigniter versi 4 dirilis sebagai lanjutan dari versi sebelumnya. Pada rilis versi ini, codeigniter menawarkan berbagai library yang dibutuhkan developer untuk memudahkan pengembangan aplikasi berbasis web. Salah satunya adalah library untuk mengakses database MySQL. Library berisi kumpulan query builder yang dapat digunakan untuk menjalankan query MySQL dengan ringkas. Penelitian ini difokuskan untuk melakukan analisis dan pengujian performansi query builder MySQL pada Codeigniter 4. Pengujian dilakukan dengan mengamati parameter Execution Time (ET) dari tiap transaksi *insert*, *update* dan *delete* pada MySQL. Hasil pengujian menunjukkan nilai rata-rata terendah dengan sample data terbesar yakni 100.000 record didapatkan pada transaksi insert dengan nilai ET 288.19 ms sedangkan nilai rata-rata tertinggi yakni pada transaksi delete dengan nilai ET 1,300 ms. Hasil penelitian ini dapat dijadikan referensi bagi pengguna framework codeigniter 4 untuk keperluan transaksi data dalam jumlah tertentu.

Kata kunci: *Performansi query, codeigniter, query builder, PHP*

Abstract— Currently the use of databases continues to increase along with the development of information technology. One of the most widely used database platforms today is MySQL. MySQL databases offer a variety of conveniences for handling various types of data storage. In addition, the MySQL database is also open source so it has many developers and has a large number of users when compared to other platforms. The use MySQL database can be managed easily using the Application Programming Interface (API) through various types of programming languages, one of which is PHP. CodeIgniter is a type of framework written in the PHP programming language. In 2020 codeigniter version 4 was released as a continuation of the previous version. In this release, codeigniter offers various libraries that developers need to make it easier to develop web-based applications. One of them is a library to access MySQL databases. The library contains a collection of query builders that can be used to run MySQL queries concisely. This research is focused on analyzing and testing the performance of MySQL query builder on Codeigniter 4. The simulation is carried out by observing the Execution Time (ET) parameter of insert, update and delete transaction in MySQL. The test results show that the lowest average value with the largest data sample is 100,000 records obtained in insert transactions with an ET value of 288.19 ms while the highest average value is in delete transactions with an ET value of 1,300 ms. The results of this study can be used as a reference for those who will use the CodeIgniter 4 framework for the purposes of data transactions in a certain amount.

Keyword: *Query performance, codeigniter, query builder, PHP*

I. PENDAHULUAN

Saat ini perkembangan teknologi informasi begitu pesat dan menyebabkan kebutuhan akan media penyimpanan data berbasis database semakin meningkat. Teknologi basis data atau sering dikenal dengan istilah *database* pun terus berkembang dan tersedia dalam berbagai platform. MySQL merupakan salah satu *platform* perangkat lunak manajemen *database* relasional yang paling banyak digunakan saat ini [1]. *Database* ini selain bersifat *open source* (kode sumbernya terbuka) juga menawarkan berbagai keunggulan untuk kebutuhan penyimpanan data dengan jenis yang beragam.

Pada implementasinya pengaksesan *database* MySQL dapat dilakukan dengan berbagai cara salah satunya adalah menggunakan *Application Programming Interface* (API) yang telah tersedia pada berbagai bahasa pemrograman seperti PHP dan yang lainnya [2]. Penggunaan API memberikan kemudahan pada proses interkoneksi berbagai *platform* aplikasi sehingga dapat menyelesaikan banyak permasalahan terkait integrasi data.

CodeIgniter adalah sebuah *framework* yang ditulis dengan bahasa pemrograman PHP. Framework ini dapat digunakan untuk memudahkan perancangan aplikasi berbasis web. Codeigniter pertama kali dirilis pada tahun 2006 dan sampai saat ini sudah memasuki versi 4 yang dirilis pada tahun 2020 sebagai lanjutan dari versi sebelumnya [3].

Pada rilis versi 4 ini, codeigniter telah banyak melakukan perubahan seperti *dukungan live web server*, instalasi *module via composer* dan beberapa keunggulan lainnya yang dapat memudahkan developer dalam pengembangan aplikasi berbasis web [4].

Query builder adalah salah satu *library* yang disediakan *framework* codeigniter untuk kemudahan pengaksesan *database*. *Library* ini berisi sekumpulan *method* atau *function* yang dapat digunakan untuk menjalankan *query* MySQL dengan ringkas. Penelitian ini berfokus untuk menganalisis dan melakukan pengujian performansi *query builder* pada codeigniter 4. Pengujian dilakukan dengan melakukan pengamatan terhadap parameter *Execution Time* (ET) untuk tiap transaksi data (*insert*, *update* dan *delete*) pada MySQL.

Nilai parameter *execution time* didapatkan dari hasil *debug* yang disediakan oleh *framework* codeigniter 4. Pengujian ini dilakukan dengan sampel data terkecil yakni 10.000 *record* dan sampai data terbesar 100.000 *record*. Masing-masing sampel data diuji untuk tiap jenis transaksi yakni *insert*, *update* dan *delete*. Hasil pengujian menunjukkan nilai rata-rata terendah untuk jumlah sampel data tertinggi yakni 100.000 *record* didapatkan untuk transaksi *insert* dengan nilai ET 288.19 ms sedangkan nilai rata-rata tertinggi diperoleh pada transaksi *delete* dengan nilai ET 1,300 ms. Hasil penelitian ini dapat menjadi referensi terhadap keperluan transaksi data dalam jumlah tertentu menggunakan *framework* codeigniter 4.

II. STUDI PUSTAKA

Pada bagian ini akan dipaparkan mengenai *database* MySQL, API, PHP, dan *framework* codeigniter.

A. MySQL

MySQL adalah perangkat lunak manajemen *database* relasional. MySQL merupakan *database open source* yang kini telah di akuisisi oleh Oracle [5]. Adapun keunggulan dari MySQL adalah sebagai berikut.

1. *Open source* dimana seluruh lisensinya berada dibawah GPL, LGPL, atau BSD.
2. Memiliki berbagai pilihan penyimpanan, baik untuk yang memerlukan penyimpanan kinerja tinggi maupun untuk keperluan bekerja dengan jenis sumber data dari *platform database* lainnya.
3. Mampu berjalan pada lintas sistem operasi dan mendukung berbagai jenis bahasa pemrograman.
4. Berbasis bahasa *query* yang standar dan mudah dipahami.
5. Sudah mendukung Galera *cluster* teknologi yang dikembangkan oleh Codership.

B. Hypertext Preprocessor (PHP)

PHP yang merupakan sebuah bahasa pemrograman web yang difungsikan untuk membangun suatu *website* dinamis dan juga bisa digunakan sebagai bahasa pemrograman umum. PHP adalah bahasa pemrograman yang paling populer digunakan saat ini [6]. PHP berjalan pada sisi server sehingga dikenal sebagai *Server Side Scripting*. Artinya setiap menjalankan PHP, wajib adanya *web server*.

C. Application Programming Interface (API)

API adalah kumpulan kode program/fungsi yang menjadi penghubung antar aplikasi baik sesama maupun beda *platform* [7]. Layanan API dapat digunakan dengan menulis sintaks/perintah tertentu melalui program yang disediakan oleh penyedia untuk dapat mengakses layanan tertentu yang diberikan. Secara lebih detil konsep API bisa dilihat pada gambar 1.



Gambar 1. Konsep API

D. CodeIgniter

CodeIgniter merupakan sebuah *framework* berbasis bahasa pemrograman PHP yang dapat digunakan untuk perancangan aplikasi berbasis web. Kehadiran *framework* ini pertama kali dirilis pada 26 februari tahun 2006 [3]. Saat ini pengembangan codeigniter sudah memasuki versi 4 yang telah dirilis pada tahun 2020 dengan membawa banya perubahan diantaranya:

1. Menggunakan PHP 7.2 ke atas dan telah mendukung PHP versi terbaru yakni PHP 8).
2. Menggunakan *Namespace* sehingga memudahkan dalam *library / script* dari pihak ketiga.
3. Memiliki fitur REST sehingga dapat langsung menggunakan REST API.
4. Memiliki CLI yang memungkinkan untuk menggunakan *task* melalui *command line*
5. Dukungan *live web server* yang memudahkan untuk dapat langsung menjalankan aplikasi tanpa perlu dukungan *web server* dari pihak ketiga.

Selain beberapa perubahan fitur di atas, *framework* ini juga membawa banyak fitur lainnya yang telah ada pada versi-versi sebelumnya seperti *library query builder*. *Library* ini dapat digunakan untuk kemudahan berkomunikasi dengan *database*. *Query builder* pada dasarnya adalah sekumpulan *method* atau *function* untuk menjalankan *query database* dengan ringkas. Artinya dengan menggunakan *library* ini penulisan *query database* dapat ditulis lebih ringkas sehingga menghemat waktu dan memperingkat kode program.

III. METODE

Pada bagian ini akan dipaparkan mengenai instalasi MySQL, codeigniter, skenario pengujian dan hasil pengujian.

A. Instalasi MySQL

Instalasi MySQL pada penelitian ini menggunakan aplikasi XAMPP. Pada aplikasi XAMPP sudah disediakan beberapa paket aplikasi seperti MySQL, Apache *web server* dan beberapa fitur yang lainnya. Berikut ini adalah tahapan-tahapan instalasi MySQL menggunakan aplikasi XAMPP.

1. Mengunduh aplikasi XAMPP dari situs www.apachefriends.org

2. Melakukan instalasi dengan pada *file* xampp yang telah didownload
3. Setelah instalasi selesai, aplikasi XAMPP sudah bisa dijalankan dengan mililih tombol *start* pada modul *service apache* dan seperti pada gambar 1.



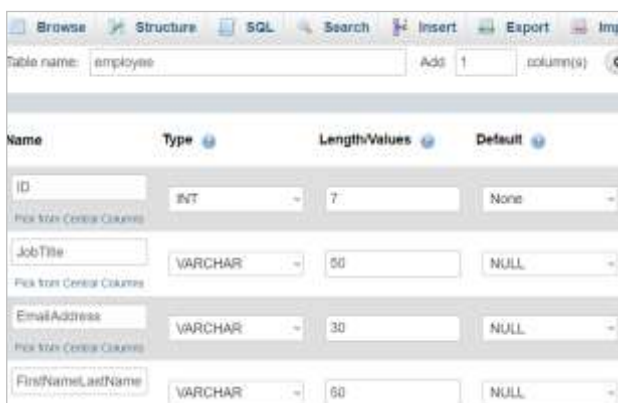
Gambar 2. Instalasi MySQL via XAMPP

4. Menyiapkan *database*. Pada tahapan ini dilakukan penyiapan *database* baru untuk keperluan pengamatan nilai ET pada masing-masing transaksi. Pembuatan *database* baru ini dilakukan melalui menu *databases*, kemudian mengisi nama *database* yang akan dibuat pada kolom *create database* seperti pada gambar 3.



Gambar 3. Membuat *database* baru pada MySQL

Setelah pembuatan *database* baru selesai, tahapan selanjutnya adalah menyiapkan tabel yang akan digunakan untuk menampung data dari tiap-tiap transaksi yang akan dilakukan. Pembuatan tabel pada *database* MySQL dapat dilakukan melalui menu *create table*. Sebagai contoh pada penelitian ini nama tabel yang dibuat adalah *employee* dengan stuktur tabelnya seperti pada gambar 4.



Gambar 4. Membuat tabel *employee* pada *database* *et_project*

B. Instalasi *framework* codeigniter 4

Untuk melakukan instalasi *framework* codeigniter 4 tahapannya sebagai berikut.

1. Menginstall *composer* terlebih dahulu dengan mengunduhnya dari situs www.getcomposer.org. *Composer* ini diperlukan untuk melakukan instalasi codeigniter pada tahapan berikutnya via *command line*.
2. Menginstall aplikasi *vscode* dengan mengunduhnya dari situs www.code.visualstudio.com. *Vscode* ini akan digunakan sebagai editor pemrograman sekaligus untuk menjalankan *composer* dan codeigniter via *command line*.
3. Mengunduh codeigniter 4 menggunakan *commang line* via *vscode* dengan memasukkan perintah *composer create-project codeigniter4/appstarter et_project -vvv*. Perintah ini digunakan untuk mengunduh *source* codeigniter terbaru kemudian menyimpannya pada sebuah *folder* dengan nama *et_project*.
4. Membuka direktori *et_project* via *vscode* yang telah diunduh pada step 3 kemudian menjalankan perintah *composer install -vvv* untuk memasang semua *library* yang dibutuhkan codeigniter 4.
5. Tahapan yang terakhir adalah menjalankan codeigniter dengan perintah *php spark serve* via *command line*. Tampilan awal *framework* codeigniter 4 dapat dilihat pada gambar 4.



Gambar 5. Codeigniter 4

C. Skenario Pengujian

Pada tahapan ini skenario pengujian dibagi menjadi 2 bagian yaitu pengujian dengan sampel 10.000 dan 100.000 *record*. Data sampel pengujian menggunakan *database* yang bersumber dari onlinedatagenerator.com (sebuah *database generator online open source*). Sebelum melakukan pengujian, variabel awal akan ditentukan seperti spesifikasi *hardware* dan *software* yang digunakan.

Adapun *hardware* yang digunakan yaitu 1 unit komputer dengan spesifikasi *processor* intel corei7-7300U dengan memori 16GB dan penyimpanan menggunakan SSD NVME 512GB. Adapun *software* yang digunakan adalah sistem operasi windows 10 Pro, XAMPP sebagai *database server* dan VSCode sebagai editor pemrograman.

1. Hasil pengujian

Hasil pengujian ET untuk masing-masing skenario pengujian dapat dijelaskan sebagai berikut:

- a) Hasil pengujian dengan sampel 10.000 *record*.

Pada pengujian ini parameter ET diamati dengan melakukan 15 kali pengujian untuk masing-masing transaksi data (*insert*, *update* dan *delete*) menggunakan sampel data 10.000 *record*. Artinya setiap proses *insert*, *update* dan *delete* dilakukan sebanyak 10.000 *record* data per tiap transaksi. Adapun hasil pengujiannya dapat dilihat pada tabel 1.

Tabel 1 ET dengan sampel data 10.000 *record*

Pengujian ke	Jenis transaksi		
	Insert	Update	Delete
1	317.17 ms	662.00 ms	172.28 ms
2	477.05 ms	619.44 ms	130.52 ms
3	361.92 ms	624.83 ms	124.97 ms
4	331.57 ms	686.90 ms	180.16 ms
5	342.70 ms	654.59 ms	158.58 ms
6	290.58 ms	684.04 ms	126.45 ms
7	283.65 ms	687.65 ms	107.75 ms
8	288.04 ms	655.96 ms	110.66 ms
9	298.11 ms	657.48 ms	190.71 ms
10	495.72 ms	627.30 ms	189.66 ms
11	295.23 ms	649.57 ms	109.56 ms
12	273.64 ms	666.29 ms	151.66 ms
13	393.34 ms	671.72 ms	102.48 ms
14	306.65 ms	680.71 ms	109.04 ms
15	286.22 ms	683.25 ms	120.64 ms
Avg	336.11 ms	660.78 ms	139.01 ms

Bersarkan tabel 1 dapat disimpulkan nilai rata-rata paling rendah didapatkan untuk jenis transaksi *insert* dengan nilai ET 139.01 ms dan nilai rata-rata tertinggi pada transaksi *update* dengan nilai ET 660.78 ms.

b) Hasil pengujian dengan sampel 100.000 *record*.

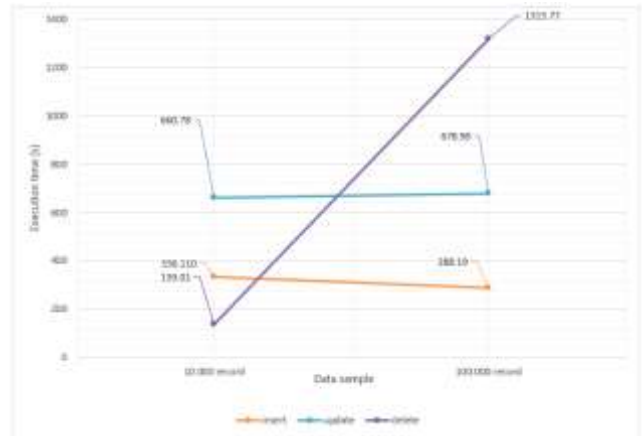
Sama seperti pengujian sebelumnya, pada pengujian ini parameter ET diamati dengan melakukan 15 kali pengujian untuk masing-masing transaksi data menggunakan sampel data 100.000 *record*. Adapun hasil pengujiannya dapat dilihat pada tabel 2.

Tabel 2 ET dengan sampel data 100.000 *record*

Pengujian ke	Jenis transaksi		
	Insert	Update	Delete
1	259.52 ms	648.21 ms	1,146.90 ms
2	293.67 ms	690.90 ms	1,406.18 ms
3	322.08 ms	666.62 ms	1,314.78 ms
4	291.56 ms	635.02 ms	1,027.17 ms
5	288.19 ms	647.65 ms	1,802.90 ms
6	279.51 ms	696.35 ms	1,778.76 ms
7	247.87 ms	657.71 ms	1,606.70 ms
8	291.24 ms	729.37 ms	1,357.42 ms
9	293.78 ms	701.66 ms	1,067.72 ms
10	285.96 ms	647.13 ms	1,060.86 ms
11	284.76 ms	654.68 ms	1,075.37 ms
12	288.57 ms	674.07 ms	1,057.79 ms
13	295.13 ms	703.76 ms	1,294.98 ms
14	304.06 ms	771.72 ms	1,322.32 ms
15	296.94 ms	658.63 ms	1,416.65 ms
Avg	288.19 ms	678.90 ms	1,315.77 ms

Bersarkan tabel 2 dapat disimpulkan nilai rata-rata paling rendah didapatkan untuk jenis transaksi *insert* dengan nilai ET 288.19 ms dan nilai rata-rata tertinggi pada transaksi *update* dengan nilai ET 678.90 ms.

Adapun visualisasi hasil pengujian dengan sampel 10.000 dan 100.000 *record* dapat dilihat pada gambar 6.



Gambar 6. Perbandingan ET

IV. KESIMPULAN

Setelah melakukan berbagai prosedur pengujian, maka dapat disimpulkan bahwa pada sampel data 10.000 *record* nilai *Executin Time* (ET) terendah didapatkan pada jenis transaksi *delete* yakni 139.01 ms. Sebaliknya pada sampel data 100.000 *record* nilai ET terbesar terjadi pada transaksi *delete* dengan nilai ET 1.3 detik. Nilai ET pada transaksi *update* cenderung menurun pada jumlah sampel 100.000 *record* yakni 288.19 ms dibandingkan dengan sampel 10.000 *record* dengan nilai ET 336.11 ms.

DAFTAR PUSTAKA

- [1] A. Helinda, Z. Musliyana, D. R. Y. Tb, M. Dwipayana, J. Suanda, and A. N. Johari, "Performance analysis of heterogeneous database management system (DBMS) synchronization using message digest 5," in *AIP Conference Proceedings*, 2020, vol. 2291, doi: 10.1063/5.0027970.
- [2] D. Mysq, "MySQL Connectors," 2022, [Online]. Available: <https://dev.mysql.com/doc/connectors/en/connectors-apis.html>.
- [3] wiki, "CodeIgniter," 2022, [Online]. Available: <https://en.wikipedia.org/wiki/CodeIgniter>.
- [4] Hadyan Palupi, "Apa Yang baru Di CodeIgniter 4," 2021, [Online]. Available: <https://codepolitan.com/blog/apa-yang-baru-di-codeigniter-4>.
- [5] I. WARMAN and R. RAMDANIANSYAH, "ANALISIS PERBANDINGAN KINERJA QUERY DATABASE MANAGEMENT SYSTEM (DBMS) ANTARA MySQL 5.7.16 DAN MARIADB 10.1," *J. TEKNOIF*, vol. 6, no. 1, 2018, doi: 10.21063/jtif.2018.v6.1.32-41.
- [6] W3Tech, "Usage of server-side programming languages for websites," *W3Tech Web Technology Surveys*, 2022.

https://w3techs.com/technologies/overview/programming_language/all.

- [7] Z. Musliyana, A. G. Satira, M. Dwipayana, and A. Helinda, "Integrated Email Management System Based Google Application Programming Interface Using OAuth 2.0 Authorization Protocol," *Elkawnie*, vol. 6, no. 1, 2020, doi: 10.22373/ekw.v6i1.5545.